REMARKS/ARGUMENTS

Claims 1-21 are pending in the present application. Reconsideration of the claims is respectfully requested.

I. Interview Summary

To be scheduled.

II. 35 U.S.C. § 102, Anticipation: Claims 1-8

The Examiner rejects claims 1-8 as being anticipated by *Bugnion*, et al., Virtual Machine Monitors for Scalable Multiprocessors, U.S. Patent 6,075,938, June 13, 2000 (hereinafter "*Bugnion*"). This rejection is respectfully traversed.

In regards to claim 1, the Examiner asserts the following:

> As per claim 1, *Bugnion* et al. teach granting, by a server partition in the logical partitioned data processing system, a logical resource to a client partition in the logical partitioned data processing system, wherein the logical resource corresponds to a physical resource - col. 4, lines 25-67; col. 6, lines 6-36; col. 8, lines 56-65; figs. 2 and *5;* col. 10, lines 14-25; col. 11, lines 20-56. mapping, by the client partition, the logical resource to the physical resource -col. 12, , lines 13-59; col. 14, line 66 to col. 15, line 9.

Office Action dated October 10, 2006, page 2.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is *identically shown* in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). In the case at hand, *Bugnion* does not anticipate the present invention as recited in claim 1 because *Bugnion* fails to teach each and every element of claim 1. Claim 1 is as follows:

> 1. A method for managing shared resources in a logical partitioned data processing system, the method comprising:
>
> granting, by a server partition in the logical partitioned data processing system, a logical resource to a client partition in the logical partitioned data processing system, wherein the logical resource corresponds to a physical resource; and
>
> mapping, by the client partition, the logical resource to the physical resource.

Bugnion does not teach the feature "granting, *by a server partition* in the logical partitioned data processing system, a logical resource *to a client partition* in the logical partitioned data processing system, wherein the logical resource corresponds to a physical resource" as recited in claim 1. The Examiner mistakenly asserts otherwise citing to numerous portions of *Bugnion*. Applicants will address each portion individually. The Examiner first cites to the following portion:

> The unique *virtual machine monitor of the present invention virtualizes all the resources of the machine*, exporting a more conventional hardware interface to the operating system. *The monitor manages all the resources* so that multiple virtual machines can coexist on the same multiprocessor. The virtual machine monitor allows multiple copies of potentially different operating systems to coexist on the multiprocessor. Some virtual machines can run commodity uniprocessor or multiprocessor operating systems, and others can run specialized operating systems fine-tuned for specific workloads. The virtual machine monitor schedules the virtual resources (processor and memory) or the virtual machines on the physical resources of the scalable multiprocessor.
>
> The unique virtual machine monitors of the present invention, in combination with commodity and specialized operating systems, form a flexible system software solution for multiprocessor machines. A large CC-NUMA multiprocessor, for example, can be configured with multiple virtual machines each running a commodity operating system such as Microsoft's Windows NT or some variant of UNIX. Each virtual machine is configured with the processor and memory resources that the operating system can effectively handle. The virtual machines communicate using standard distributed protocols to export the image of a cluster of machines.
>
> Although the system looks like a cluster of loosely-coupled machines, *the virtual machine monitor uses global policies to manage all the resources of the machine*, allowing workloads to exploit the fine-grain resource sharing potential of the hardware. For example, the monitor can move memory between virtual machines to keep applications from paging to disk when free memory is available in the machine. Similarly, the monitor dynamically schedules virtual processors on the physical processors to balance the load across the machine. The use of commodity software leverages the significant engineering effort invested in these operating systems and allows CC-NUMA machines to support their large application base. Since the monitor is a relatively simple piece of code compared to large operating systems, this can be done with a small implementation effort as well as with a low risk of introducing software bugs and incompatibilities.

Bugnion, col. 4, lines 25-67 (Emphasis added).

The above portion of *Bugnion* teaches a virtual machine monitor that virtualizes all the resources of the machine so that multiple virtual machines can coexist on the same multiprocessor. The virtual machine monitor schedules the virtual resources (processor and memory) or the virtual machines on the physical resources of the scalable multiprocessor. The virtual machine monitor uses global policies to manage all the resources of the machine, allowing workloads to exploit the fine-grain resource sharing potential of the hardware.

The above portion fails to teach the feature "granting, *by a server partition* in the logical partitioned data processing system, a logical resource *to a client partition* in the logical partitioned data processing system, wherein the logical resource corresponds to a physical resource" as recited in claim 1. As stated in *Bugnion*, the *virtual machine monitor* manages all the resources of the machine. The examples given in the above portion state that "*the monitor* can move memory between virtual machines" and "*the monitor* dynamically schedules virtual processors on the physical processors to balance the load across the machine. Even if, *arguendo*, a virtual machine as described in *Bugnion* is the same as a partition as described in the presently claimed invention, *Bugnion* does not teach a virtual machine (i.e. *a server partition*) granting a logical resource to another virtual machine (i.e. *a client partition*). Instead, *Bugnion* teaches a 3rd party component, a *virtual machine monitor*, that manages all the resources of the machine. Contrary to *Bugnion*, claim 1 recites that the act of granting is performed *by the server partition* and not by a 3rd party component. Therefore, the above portion of *Bugnion* does not anticipate claim 1.

The Examiner additionally cites to the following portion of *Bugnion*:

> In one aspect of the invention, a computational system is provided that comprises a multiprocessor hardware layer, a virtual machine monitor layer, and a plurality of operating systems. The multiprocessor hardware layer comprises a plurality of computer processors, a plurality of physical resources associated with the processors, and an interconnect providing mutual communication between the processors and resources. *The virtual machine monitor (VMM) layer executes directly on the hardware layer* and comprises a resource manager that manages the physical resources of the multiprocessor, a processor manager that manages the computer processors, and a hardware emulator that creates and manages a plurality of virtual machines. The operating systems execute on the plurality of virtual machines and transparently share the plurality of computer processors and physical resources *through the VMM layer.* In a preferred embodiment, *the VMM layer* further comprises a virtual network device providing communication between the operating systems executing on the virtual machines, and allowing for transparent sharing optimizations between a sender operating system and a receiver operating system. In addition, the resource manager maintains a global buffer cache that is transparently shared among the virtual machines using read-only mappings in portions of an address space of the virtual machines. The VMM layer also maintains copy-on-write disks that allow virtual machines to transparently share main memory resources and disk storage resources, and performs dynamic page migration/replication that hides distributed characteristics of the physical memory resources from the operating systems. The VMM layer may also comprise a virtual memory resource interface to allow processes running on multiple virtual machines to share memory.

Bugnion, col. 6, lines 6-36 (Emphasis added).

Again, the above portion of *Bugnion* describes a virtual machine monitor (VMM) that executes directly on the hardware layer. The sharing of physical resources, processor and memory, between the operating systems executing on the plurality of virtual machines is done *through the VMM layer.* Similar to the above argument, *Bugnion* does not teach "granting, *by a server partition* in the logical partitioned data processing system, a logical resource *to a client partition* in the logical partitioned data processing system" as recited in claim 1. Instead, *Bugnion* teaches that sharing of physical resources between virtual machines is controlled by a virtual machine monitor. Therefore, the above portion of *Bugnion* does not anticipate claim 1.

The Examiner also cites to the following portion of *Bugnion*:

> FIG. 1 shows how *the virtual machine monitor* allows multiple copies of potentially different operating systems to coexist. In this figure, five virtual machines coexist on the multiprocessor. Some virtual machines run commodity uniprocessor or multiprocessor operating systems, and others run specialized operating systems fine-tuned for specific workloads. *The virtual machine monitor* schedules the virtual resources (processor and memory) or the virtual machines on the physical resources of the scalable multiprocessor.

Bugnion, col. 8, lines 56-65 (Emphasis added).

The above portion states *the virtual machine monitor* allows multiple copies of potentially different operating systems to coexist. Furthermore, *the virtual machine monitor* schedules the virtual resources (processor and memory) or the virtual machines on the physical resources of the scalable multiprocessor. For the same reasons as stated above, *Bugnion* does not teach "granting, *by a server partition* in the logical partitioned data processing system, a logical resource *to a client partition* in the logical partitioned data processing system." Therefore, the above portion of *Bugnion* does not anticipate claim 1.

Additionally, the Examiner cites to Figures 2 and 5. Figures 2 and 5 are as follows:

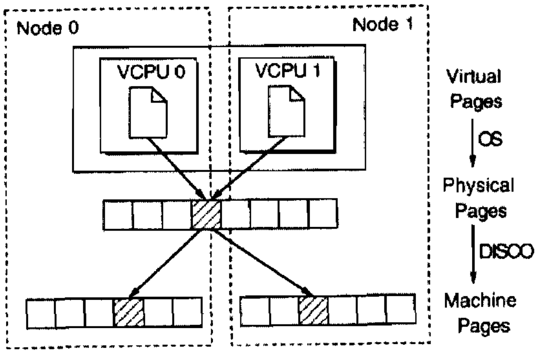
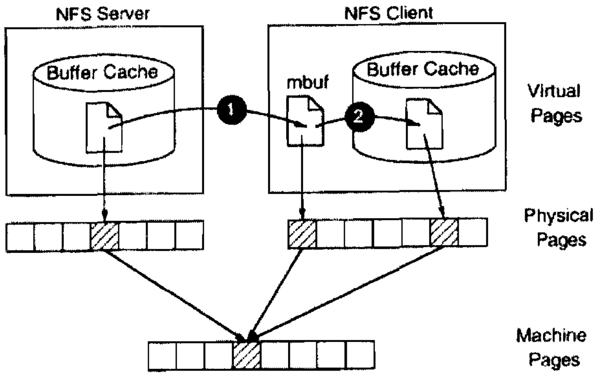


FIG. 2

FIG. 5



Bugnion, Figures 2 and 5.

Figure 2 shows Disco, a virtual machine monitor (col. 7, lines 54-55), mapping the same physical page from virtual pages from VCPUs 0 and 1. Disco transparently maps each virtual page to a machine page replica that is located on the local node (col. 7, lines 59-65).

Burgeon further describes Figure 5 as follows:

> FIG. 5 is a schematic diagram illustrating an example of transparent sharing of pages over NFS according to the present invention. The diagram shows the case when the NFS reply, to a read request, includes a data page. (1) *The monitor's* networking device remaps the data page from the source's machine address space to the destinations (2) *The monitor* remaps the data page from the driver's mbuf to the client's buffer cache. This remap is initiated by the operating system *through a monitor call.*

Bugnion, col. 8, lines 8-16 (Emphasis added).

Both Figures 2 and 5, further illustrates that the virtual machine monitor controls the sharing of physical resources among the virtual machines. However, claim 1 recites a different method from that taught in *Bugnion* because claim 1 states that the act of granting is performed *by a server partition* in the logical partitioned data processing system, a logical resource *to a client partition* in the logical partitioned data processing system. Contrary to *Bugnion*, claim 1 does not recite the use of a 3rd party component such as a virtual machine monitor to share resources. In the claimed invention, the logical resource is granted *by* the server partition and not by another 3rd party component. Consequently, *Bugnion* does not anticipate claim 1.

The Examiner further cites to the following portion:

> To support efficient communication between virtual machines, as well as other real machines, *the monitor* virtualizes access to the networking devices of the underlying system. Each virtual machine is assigned a distinct link-level address on an internal virtual subnet handled by Disco. Besides the standard network interfaces such as Ethernet and FDDI, Disco supports a special network interface that can handle large transfer sizes without fragmentation. For communication with the world outside the machine, Disco acts as a gateway that uses the network interfaces of the machine to send and receive packets.

The above portion states that Disco, the virtual machine monitor, communicates between virtual machines by virtualizing access to the network devices. The above portion does not teach granting, by a server partition a logical resource to a client partition. Therefore, for the same reasons as previously described, the above portion does not anticipate claim 1.

Lastly, the Examiner cites to the following portion:

> On the MIPS processor, Disco runs in kernel mode with full access to the machine's hardware. When control is given to a virtual machine to run, Disco puts the processor in supervisor mode if running the virtual machine's operating system, and in user mode otherwise. Supervisor mode allows the operating system to use a protected portion of the address space (the supervisor segment) but does not give access to privileged instructions or physical memory. Applications and kernel code can however still be directly executed since Disco emulates the operations that cannot be issued in supervisor mode. When a trap such as page fault, system call, or bus error occurs, the processor traps to the monitor that emulates the effect of the trap on the currently scheduled virtual processor. This is done by updating the privileged registers of the virtual processor and jumping to the virtual machine's trap vector.

> Disco maintains all the privileged registers in the VCPU structure. Privileged instructions that change the state of privileged registers are emulated by the monitor. The monitor updates the privileged registers as dictated by the instruction.
>
> In order to emulate the privileged instructions that modify and query the state of the TLB, Disco maintains a software TLB. This structure holds the untranslated TLB entries that the kernel has entered in the TLB. In order to make lookup in the software TLB fast, Disco implements a direct mapped TLB for the random entries.
>
> Hardware interrupts are handled directly by the VMM through its own device drivers. The VMM posts an interrupt to the virtual machine when the operation that it has requested completes. The mechanism that posts an interrupt to a vCPU of a VM must execute on the physical processor that currently runs that VCPU (if any) since it alters the state of the VCPU. The altered state includes the status and cause registers, as well as the exception program counter (EPC). The PC is set to the start of the VM's

Bugnion, col. 11, lines 20-56.

The above portion of *Bugnion* describes that the virtual machine monitor, Disco, can operate on a MIPS processor. Disco emulates the operations that cannot be issued in supervisor mode. Hardware interrupts are handled directly by the VMM through its own device drivers. However, the above portion does not teach granting, *by a server partition* a logical resource to a client partition. Therefore, for the same reasons as previously described, the above portion does not anticipate claim 1.

Applicants have shown that *Bugnion* does not teach the features as recited in claim 1. Therefore, *Bugnion* does not anticipate claim 1. Consequently, the rejection of claim 1 under 35 U.S.C. § 102 has been overcome.

Because claims 2-8 depends from claim 1, the same distinctions between *Bugnion* vis-à-vis claim 1 apply to claims 2-8. Additionally, claims 2-8 recite other additional combinations of features not taught by the reference. For example, claim 5 recites returning, *by the client partition*, the logical resource to the server partition. The Examiner mistakenly asserts that *Bugnion* teaches this feature. The Examiner cites to the following:

> The second example shows the impact of a page migration action. The hardware of the FLASH machine determines that a given machine page is "hot" and *Disco determines* that it is suitable for migration. The transparent migration requires that all mappings that point to that page be removed from all processors. The entry in the memmap of that machine address contains the list of the pmap entries that refer to the page. The pmap entry contains a backmap to the virtual address and a bitmask of vcpus that possibly have the mapping to that machine address. Finally, all matching entries in the relevant 12tlbs and R10000 TLBs are invalidated before the page is actually migrated.

Bugnion, col. 14, lines 19-30 (Emphasis added).

The above portion teaches that Disco determines the suitability of moving a machine page. This determination requires all mappings that point to that page be removed from all processors. Disco performs this action by invalidating all matching entries in the relevant 12tlbs and R10000 TLBs before the page is actually migrated. However, the above portion does not teach returning, *by the client partition*, the logical resource to the server partition. Therefore, *Bugnion* does not teach the additional features recited in claim 5. For the same reasons, *Bugnion* does not teach the additional feature "*rescinding, by the server partition*, the logical resource" as recited in claim 6. Thus, *Bugnion* does not anticipate claims 5 or 6.

III. 35 U.S.C. § 103, Asserted Obviousness Claims 9 and 10

The Examiner rejects claims 9 and 10 as obvious over *Bugnion* in view of *Blumenau*, et al., Mapping of Hosts to Logical Storage Units and Data Storage Ports in a Data Processing System, U.S. Publication 20040054866, March 18, 2004 (hereinafter "*Blumenau*"). This rejection is respectfully traversed.

The Examiner asserts the following:

> As per claims 9-10, *Bugnion* et al. teach responsive to a failure of the server partition, notifying the client partition of the failure of the server partition - col. 5, lines 40-47. However, *Bugnion* et al. do not teach recovering outstanding shared logical resources; for the server partition; restarting the server partition. *Blumenau* et al. teach storage: volume partitioning by named groups - pars. 0084-0086; mapping of logical volumes and physical volumes - pars. 0148, 0238; higher mean time before failure and recovering logical resources and restarting/booting the volume or partition - pars. 0062, 0170-0171. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine *Bugnion* et al. and *Blumenau* et al. to better recovering the shared logical resources thus, allow better data sharing for users.

Office Action dated October 10, 2006, page 4.

III.A. The Examiner Has Failed to Present a Prima Facie Case of Obviousness Because the References Do Not Teach or Suggest All the Features of Claim 9.

Regarding claim 9, the Examiner has failed to state a *prima facie* obviousness rejection because the proposed combination does not teach or suggest all of the features of claim 9. A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). In the case at hand, not all of the features of the claimed invention have been properly considered and the teachings of the references themselves do not teach or suggest the claimed subject matter to a person of ordinary skill in the art.

Claim 9 is as follows:

> 9. The method of claim 1, further comprising:
> responsive to a failure of the server partition, notifying the client partition of the failure of the server partition;
> recovering outstanding shared logical resources for the server partition; and
> restarting the server partition.

Addressing the rejection of claim 9, the Examiner has failed to state a *prima facie* obviousness rejection because neither *Bugnion* nor *Blumenau* teach or suggest all features of claim 1, from which claim 9 depends. As discussed above, *Bugnion* does not teach the claimed feature "granting, *by a server partition* in the logical partitioned data processing system, a logical resource *to a client partition* in the logical partitioned data processing system, wherein the logical resource corresponds to a physical resource" as recited in claim 1. Because *Bugnion* specifically teaches the use of a virtual machine monitor to manage all the resources of the machine, *Bugnion* further does not suggest the features of claim 1.

Additionally, *Blumenau* also does not teach or suggest all of the features of claim 1. *Blumenau* is directed towards an apparatus containing host ports for coupling hosts to data storage devices. The apparatus decodes a host identifier and a logical storage unit specification from each data access request received at each host port, and determines whether or not the decoded host identifier and logical storage unit specification are in conformance with the mapping in order to permit or deny data access of the logical storage unit through the host port (*Blumenau,* Abstract). *Blumenau* does not teach the claimed feature "granting, *by a server partition* in the logical partitioned data processing system, a logical resource *to a client partition* in the logical partitioned data processing system, wherein the logical resource corresponds to a physical resource" as recited in claim 1. Furthermore, *Blumenau is* not directed towards the sharing of logical resources between server and client partitions, therefore, *Blumenau* also does not suggest the features of claim 1.

Because neither *Bugnion* nor *Blumenau* teach or suggest all of the features of claim 1, and because claim 9 depends from claim 1, the proposed combination of *Bugnion* and *Blumenau* when considered as a whole does not teach or suggest all of the features of claim 9. Accordingly, the Examiner has failed to state a *prima facie* obviousness rejection of claim 9.

Additionally, the proposed combination of *Bugnion* and *Blumenau,* when considered as a whole, also does not teach or suggest all of the features of claim 9. Claim 9 recites recovering outstanding shared logical resources for the server partition responsive to a failure of the server partition. The Examiner admits that *Bugnion* does not teach this feature. Additionally, *Bugnion* provides no reference to suggest this feature.

The Examiner erroneously asserts that *Blumenau* teaches this additional feature. The Examiner cites to the following portions of *Blumenau:*

> The storage subsystem 20 is constructed for high data availability so that a single high-capacity storage subsystem is at least as fault-tolerant as a local collection of conventional network storage servers. Fault tolerance is ensured by dual, redundant components and busses in the path from any one of the port adapters 35, 36 to any one of the storage devices 28, 29, 30, and 31. Mirroring or RAID (redundant array of inexpensive disks) techniques ensure that the storage adapters 37, 38 *can recover data* in the event of failure of any one of the storage devices. In a similar fashion, the data network 21 can be made fault tolerant by ensuring that each of the hosts 22, 23, 24, 25 has independent paths through the data network 21 to each of two of the port adapters 35, 36, as will be further described below with reference to FIG. 2.

Blumenau, par. 0062 (Emphasis added).

The above portion teaches providing fault-tolerant control of a storage subsystem. Fault tolerance is ensured by dual, redundant components. This teaching allows for the recovery of data in the event of failure of any one of the storage devices. *Blumenau* provides a method for the recovery of data and not logical resources as recited in claim 1. As stated in claim 1, a logical resource corresponds to a physical resource, whereas data refers to information that is stored. Therefore, the above portion does not teach or suggest the features of claim 9.

Because neither *Bugnion* nor *Blumenau teach* or suggest the claimed feature of recovering outstanding shared logical resources for the server partition, the proposed combination when considered as a whole does not teach or suggest all of the features of claim 9. Thus, the Examiner failed to state a *prima facie* obviousness rejection of claim 9. For the same reasons the Examiner failed to state a *prima facie* obviousness rejection of claim 10.

III.B. <u>The Examiner Has Failed to Present a Prima Facie Case of Obviousness Because the References Address Different Problems, So No Teaching, Suggestion, or Motivation Exists to Combine the References in the Suggested Manner</u>.

A *prima facie* obviousness rejection against claims 9 and 10 has not been made because no proper teaching or suggestion to combine the references has been stated. A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). A proper *prima facie* case of obviousness cannot be established by combining the teachings of the prior art absent some teaching, incentive, or suggestion supporting the combination. *In re Napier*, 55 F.3d 610, 613, 34 U.S.P.Q.2d 1782, 1784 (Fed. Cir. 1995); *In re Bond*, 910 F.2d 831, 834, 15 U.S.P.Q.2d 1566, 1568 (Fed. Cir. 1990). No such teaching or suggestion is present in the cited references.

The references themselves do not suggest the proposed combination. Furthermore, it would not have been obvious to one of ordinary skill in the art, at the time the invention was made, to combine the teachings of *Bugnion* and *Blumenau* because both references seek to provide solutions to dissimilar problems. *Bugnion* teaches a virtual machine monitor that allows multiple virtual machines to coexist and share physical resources. The virtual machine monitor manages all the resources of the machine. On the other hand, *Blumenau* is not related to the sharing of physical resources among multiple virtual machines. *Blumenau* is directed towards an apparatus for determining whether or not to permit or deny data access of the logical storage unit through the host port. The problem *Bugnion* seeks to solve, controlling physical resources among virtual machines, is unrelated to the problem of determining access to a logical storage unit which *Blumenau* seeks to provide a solution. Therefore, one of ordinary skill would not combine the teachings of *Bugnion* and *Blumenau* to reach the presently claimed invention. Accordingly, no prima facie obviousness rejection has been stated against the claims.

Claims 11-21

The Examiner rejects claims 11-21 under the same rationale as claim 1-10. Therefore, Applicants have overcome the rejection against these claims at least for the reasons provided vis-à-vis the response to the rejection of claim 1-10.

IV. Conclusion

The subject application is patentable over the cited references and should now be in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: January 10, 2007

Respectfully submitted,

/Theodore D. Fay III/

Theodore D. Fay III
Reg. No. 48,504
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

TF/nh